

## Examen de remplacement – Informatique 2

### Questions de cours : [4 points]

- 1/ Quelle est la différence entre un paramètre passé par variable et un paramètre passé par valeur ? (1 point)
- 2/ Comment appelle-t-on les variables de type tableau ? (1 point)
- 3/ Quelle est la principale différence entre une procédure et une fonction ? (1 point)
- 4/ Donner la syntaxe de la déclaration d'un enregistrement. (1 point)

### Exercice 01 : [8 points]

Soit A une matrice carrée de  $N \times N$  composantes entières et V un vecteur de N composantes entières (avec  $3 \leq N \leq 8$ ) :

Partie A : Écrire un programme pascal qui permet de :

- 1- Trouver le nombre des éléments pairs dans le vecteur V.
- 2- Trouver la plus grande valeur paire dans le vecteur V et sa position.

**N.B** : La première valeur du vecteur V est considérée comme élément pair.

Partie B : Modifier le programme afin de :

- 1- Calculer le nombre d'éléments non nuls de l'anti-diagonale de la matrice A.
- 2- Remplacer la première ligne de la matrice A par les éléments du vecteur V.

**N.B** : L'entrée N doit être contrôlée.

### Exercice 02 : [8 points]

Soit le programme Pascal suivant :

```
Program pgcd_ppcm;
var
  x, y, ppcm : integer;
  ..... calculer_pgcd(a, b: integer): integer;
var
  r: integer;
begin
  while b <> 0 do
    begin
      r := a mod b;
      a := b;
      b := r;
    end;
    ..... := a;
end;
..... calculer_ppcm(a, b: integer ; var ppcm: integer);
var
  pgcd : integer;
begin
  pgcd := calculer_pgcd(x, y);
  ppcm := (a * b) div pgcd;
end;
BEGIN {Début du programme principal}
  readln(x, y);
  calculer_ppcm(x, y, ppcm);
  writeln('Le PPCM de ', x, ' et ', y, ' est ', ppcm);
END. {Fin du programme principal}
```

### Questions :

1. Identifier les types des sous-programmes présents dans ce programme. (0.5 point)
2. Compléter les sous-programmes avec ce qui manque. (1.5 point)
3. Quels sont les paramètres formels des sous-programmes donnés ? (1.5 point)
4. Dérouler le programme pour X=6 et Y=2. (2.5 points)
5. Réécrire le programme en transformant les deux sous-programmes donnés à l'autre type de sous-programme. (2 points)

## Corrigé de l'examen de remplacement – Informatique 2

### Questions de cours :

- 1- Quelle est la différence entre un paramètre passé par variable et un paramètre passé par valeur ?

**Réponse :** Dans le **passage par valeur**, une copie des paramètres effectifs est transmise aux paramètres formels respectifs. Alors que, dans le **passage par variable**, l'emplacement (l'adresse mémoire) des paramètres effectifs est transmis à des paramètres formels, toute modification apportée aux paramètres formels se reflétera également dans les paramètres effectifs. (Ou bien l'utilisation du mot clé Var avant la déclaration du paramètre formel)

1 pt

- 2- Comment appelle-t-on les variables de type tableau ?

**Réponse :** Les tableaux sont appelés les variables indicées.

1 pt

- 3- Quelle est la principale différence entre une procédure et une fonction ?

**Réponse :** Une fonction possède toujours un type de retour.

1 pt

- 4- Donner la syntaxe de la déclaration d'un enregistrement.

**Réponse :** La syntaxe pour déclarer un enregistrement :

1 pt

**Type** <id\_enregistrement> = **record** ;

< Champ 1 > : <Type 1> ;

< Champ 2 > : <Type 2> ;

< Champ N > : <Type N> ;

**End;**

### Exercice 01 : [8 points]

Soit A une matrice carrée de  $N \times N$  composantes entières et V un vecteur de N composantes entières (avec  $3 \leq N \leq 8$ ) : **(1,5 points pour la déclaration des variables et 1 point pour le contrôle de la valeur N et 1 points pour l'initialisation du vecteur et la matrice)**

Partie A : Écrire un programme pascal qui permet de :

- 1- Trouver le nombre des éléments pairs dans le vecteur V. **(1 point)**
- 2- Trouver la plus grande valeur paire et la plus petite valeur impaire dans le vecteur V et leurs indices (positions). **(1,125 point)**

Partie B : Modifier le programme afin de :

- 1- Calculer le nombre d'éléments non nuls de l'anti-diagonale de la matrice A. **(1 point)**
- 2- Remplacer la première ligne de la matrice A par les éléments du vecteur V. **(1,5 point)**

**N.B** : L'entrée N doit être contrôlée.

#### Programme Pascal

```
Program exo1;
Var
A : array [1..100,1..100] of integer; (0,25)
V : array [1..100] of integer; (0,25)
N, i, j, cp, maxp, posmax, nbrnul : integer; (0,125*8)
} 1,5

Begin
Repeat (0,25)
Writeln ('Donner la valeur de N:');
Read (N); (0,25)
Until ( (N >= 1) and (N <= 8) ); (0,5)
} 1

Writeln ('Donner les composantes de la matrice A:');
For i := 1 to N Do (0,125)
  For j := 1 to N Do (0,125)
    Read (A[i,j]); (0,25)
} 0,5

Writeln ('Donner les composantes du vecteur V:');
For i := 1 to N Do (0,25)
  Read (V[i]); (0,25)
} 0,5

cp := 0; (0,25)
For i := 1 to n do (0,125)
Begin
If (V[i] mod 2 = 0) then (0,125)
cp := cp + 1 ; (0,25)
End;
Writeln ('le nombre des elements pairs dans le vecteur est :', cp); (0,25)
} 1
```

```
maxp := v[1]; (0,125)
posmax := 1; (0,125)

For i := 1 to N Do (0,125)
begin
If ((v[i] > maxp) and (v[i] mod 2 = 0) ) then (0,25)
Begin
maxp := v[i]; (0,125)
posmax := i; (0,125)
End;
End;
Writeln ('le maximum pair est : ', maxp , ' il se trouve dans la position : ', posmax); (0,125)
```

```
nbrnul := 0; (0,125)
For i := 1 to N Do (0,125)
  if (A[i,N-i+1] <> 0) then (0,25)
    nbrnul := nbrnul + 1; (0,25)

Writeln ('Le nombre des elements non nuls dans l anti-diagonale est :', nbrnul); (0,25)
```

```
For j := 1 to N do (0,25)
  A[1,j] := V[j]; (0,5)

For i := 1 to N Do (0,25)
  Begin
    for j := 1 to N Do (0,25)
      Write (A[i,j], ' '); (0,25)
  Writeln;
  End;
```

**End.**

## Exercice 02 : [8 points]

Soit le programme Pascal suivant :

```
Program pgcd_ppcm;
var
  x, y, ppcm : integer;
  ..... calculer_pgcd(a, b: integer): integer;
var
  r: integer;
begin
  while b <> 0 do
    begin
      r := a mod b;
      a := b;
      b := r;
    end;
    ..... := a;
  end;
  ..... calculer_ppcm(a, b: integer ; var ppcm: integer);
var
  pgcd : integer;
begin
  pgcd := calculer_pgcd(x, y);
  ppcm := (a * b) div pgcd;
end;
BEGIN
  readln(x, y);
  calculer_ppcm(x, y, ppcm);
  writeln('Le PPCM de ', x, ' et ', y, ' est ', ppcm);
END.
```

### Questions :

1. Identifier les types des sous-programmes présents dans ce programme. **(0.5 point)**
2. Compléter les sous-programmes avec ce qui manque. **(1.5 point)**
3. Quels sont les paramètres formels des sous-programmes donnés ? **(1.5 point)**
4. Dérouler le programme pour X=6 et Y=2. **(2.5 points)**
5. Réécrire le programme en transformant les deux sous-programmes donnés à l'autre type de sous-programme. **(2 points)**

### Réponse :

1. Le type des sous-programmes donnés sont : **(0.5 point)**

Le sous-programme 1 : **fonction**

0.25 pt

Le sous-programme 2 : **procédure**

0.25 pt

## 2. Compléter les sous-programmes avec ce qui manque. (1.5 point)

```

Program pgcd_ppcm;
var
  x, y, ppcm : integer;
Function calculer_pgcd(a, b: integer): integer;
var
  r: integer;
begin
  while b <> 0 do
    begin
      r := a mod b;
      a := b;
      b := r;
    end;
    calculer_pgcd := a;
  end;
procedure calculer_ppcm(a, b: integer ; var ppcm: integer);
var
  pgcd : integer;
begin
  pgcd := calculer_pgcd(x, y);
  ppcm := (a * b) div pgcd;
end;
BEGIN
  readln(x, y);
  calculer_ppcm(x, y, ppcm);
  writeln('Le PPCM de ', x, ' et ', y, ' est ', ppcm);
END.

```

0.5 pt

0.5 pt

0.5 pt

## 3. Les paramètres formels des sous-programmes donnés . (1.5 point)

Les paramètres formels des deux sous-programmes :

Type de paramètres\Sous-programme	Sous_prog1	Sous_prog2
Paramètres formels	a, b, calculer_pgcd	a, b et ppcm

0.25 pt x 3

0.25 pt x 3

4. Dérouler le programme pour X=6 et Y=2. (2.5 points)

Instructions	P.P			SP1 (fonction)				SP2 (procédure)			Affichage
	x	y	ppcm	a	b	r	calculer_pgcd	a	b	ppcm	
readln(x, y);	6	2									0.25
calculer_ppcm(x, y, ppcm);			Ppcm=?								0.25
//Appel à la procédure <b>calculer_ppcm</b> avec x=6 et y=2											0.125
//Transmission des paramètres								6	2		0.25
pgcd := calculer_pgcd(x, y);											0.25
//Appel à la fonction <b>calculer_pgcd</b> avec x=6 et y=2											0.125
//Transmission des paramètres				6	2						0.25
while b <> 0 do 2 <> 0 <b>vrai</b>											0.25
On rentre dans la boucle while											0.5
r := a mod b; <b>6 mod 2 = 0</b>						0					0.125
a := b; a:=2				2							0.125
b := r; <b>b:=0</b>					0						0.125
<b>calculer_pgcd</b> := a; <b>calculer_pgcd</b> := 2;							2				0.25
//Retour du résultat dans la fonction <b>calculer_pgcd</b>											0.25
ppcm := (a * b) div pgcd;											0.125
<b>ppcm := 12 div 2 :=6</b>										6	0.125
//Retour du résultat dans le programme principale			Ppcm=6								0.125
writeln('Le PPCM de ', x, ' et ', y, ' est ', ppcm);											Le PPCM de 6 et 2 est 6 0.125

5. Réécrire le programme en transformant les deux sous-programmes donnés à l'autre type de sous-programme. (2 points)

```
Program pgcd_ppcm;
var
  x, y, ppcm : integer;
procedure calculer_pgcd(a, b: integer; var p: integer);
  0.125                                0.125                                0.125 //Suppression du type de la fonction
var
  r: integer;
begin
  while b <> 0 do
    begin
      r := a mod b;
      a := b;
      b := r;
    end;
    p := a;
    0.25                                0.25 //Suppression du type de la fonction
end;
function calculer_ppcm(a, b: integer): integer;
  0.125                                0.125                                0.125 //Suppression de la variable de sortie
var
  pgcd: integer;
begin
  calculer_pgcd(x, y, pgcd);          0.25
  calculer_ppcm := (a * b) div pgcd;  0.25
end;
BEGIN
  readln(x, y);
  ppcm := calculer_ppcm(x, y);      0.25
  writeln('Le PPCM de ', x, ' et ', y, ' est ', ppcm);
END.
```